

EV316937141

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**Automatic Baud Rate Detection of Null Modem
Unimodem Client Connection**

Inventor(s):
Kent D. Cedola

ATTORNEY'S DOCKET NO. MS1-400USC1

RELATED APPLICATIONS

This application is a continuation of U.S. Application Serial No. 09/507,480, filed February 17, 2000 and entitled "Automatic Baud Rate Detection of Null Modem Unimodem Client Connection", incorporated by reference herein for all that it discloses and teaches.

TECHNICAL FIELD

This invention relates to computer operating systems, and particularly, to computer methods for detecting baud rates for communicating with client devices over a serial null modem connection that uses a Unimodem null serial protocol.

BACKGROUND

Handheld computing devices have grown rapidly in popularity in recent years. These devices enable users to port many of the tools and features of their desktop computers on excursions away from home or the office. While these devices offer tremendous advantages for portability, most mobile users still rely on desktop computers when working at home or the office.

As a result of operating multiple computers, users often enter different information into different computers depending upon their situation and location. For instance, a user may enter tasks or other data into a work computer, appointments and contact information into a portable computer while on the road, and calendar events into a home computer when at home. Yet, when the user wants to see their schedule, the user wants to view the most up-to-date schedule, regardless of where the information was entered or where the user accesses it.

1 To avoid making the user enter the same data multiple times into different
2 computers, most handheld computers are configured to swap information with
3 desktop computers. Programs executing on one or both computers synchronize the
4 exchange of schedules, appointments, and other data contents so that any unique
5 information currently residing on one of the computers is shared with the other to
6 bring both computers up to-date.

7 Typically, handheld computers connect to a host desktop computer via a
8 serial connection. One common serial connection is known as a "null modem
9 connection", which uses a Unimodem null serial protocol to exchange data.
10 According to this protocol, the handheld computer initiates communication by
11 sending a message called "CLIENT". The host computer replies with a message
12 "S", "E", "R", "V", "E", "R", "C", "L", "I", "E", "N", "T". If both sides
13 understand the transmission, a synchronization session is begun to synchronize the
14 contents of the two computers.

15 One problem surrounding this serial protocol is that the message exchange
16 is baud rate dependent. That is, both the handheld computer and the host computer
17 must be operating at the same baud rate for each to understand the other's
18 message. However, there are many different baud rates at which the two
19 computers may communicate. For example, handheld computers that run the
20 Windows CE operating system from Microsoft Corporation can support four
21 different baud rates: 19.2K, 38K, 56K, and 115K. If the handheld computer is
22 operating at one baud rate (e.g., 19.2K) and the host computer is operating at
23 another baud rate (e.g., 56K), the two computers will not be able to communicate
24 with one another. Unfortunately, connection failures due to serial baud rate
25 mismatches tend to be common.

1 When a baud rate mismatch occurs, the user typically tries to adjust the
2 baud rate of one of the computers. It tends to be harder to change the baud rate on
3 the host computer, so the user generally ends up manually reconfiguring the
4 handheld computer to a different baud rate.

5 Another problem is that an increasing number of users own more than one
6 handheld computer. In some cases, the handheld computers may be set to different
7 baud rates, whereas the host computer can be set to only one of these baud rates at
8 a given time. As a result, the user is forced to adjust baud rates more often to
9 accommodate the multiple portable devices.

10 Thus, there is a need for an improved system and method for connecting
11 handheld computers to host computers over a null modem unimodem client
12 connection.

13 14 **SUMMARY**

15 This invention concerns a baud rate detection system and method for
16 automatically detecting the baud rate at which a client computing device is
17 communicating with a host computer over a serial connection.

18 In one implementation, the host computer is coupled to the client computing
19 device via a serial connection, which employs a Unimodem null serial protocol to
20 exchange data. The baud rate detection system includes a baud rate selector to
21 select among multiple baud rates that the client computing device might use to
22 transmit a predefined message (e.g., a text string "C", "L", "I", "E", "N", "T").
23 The system also includes a message detector to listen at the currently selected baud
24 rate for the predefined message.

1 During operation, after the client connects to the host computer, the baud
2 rate selector chooses a first baud rate from among the multiple possible baud rates.
3 The message detector listens at this baud rate for the message. If the message
4 detector receives the message, the current baud rate is the correct rate and is used
5 for continuing communications with the client computing device. On the other
6 hand, if the message detector fails to detect the message after a predetermined time
7 period or detects characters not included in the predefined message, the baud rate
8 selector chooses a new baud rate and the message detector begins listening at the
9 new baud rate. This process continues until the baud rate detection system finds
10 the appropriate baud rate for communicating with the client computing device.

11 The automatic baud rate detection is advantageous because it effectively
12 eliminates connection failures caused by baud rate mismatches. This obviates the
13 need for a user to manually reconfigure either the host or client computing device
14 when baud rate mismatches occur.

15 16 **BRIEF DESCRIPTION OF THE DRAWINGS**

17 Fig. 1 is a block diagram of a host computer connected to a client
18 computing device, such as a handheld computer.

19 Fig. 2 is a flow diagram of a method for automatically detecting a baud rate
20 at which the client computing device is transmitting data.

21 22 **DETAILED DESCRIPTION**

23 Fig. 1 shows a computer system 20 having a host computer 22 connected to
24 a client computing device 24 via a serial connection 26. The host computer 22
25 may be embodied in many ways including, for example, a workstation, a desktop

1 computer, a laptop computer, or the like. The client computing device 24 may also
2 be implemented in a number of ways, such as a handheld computer, a telephone or
3 other communication device, a personal digital assistant, and so forth. The serial
4 connection 26 is preferably employs a null modem Unimodem client protocol,
5 which establishes a connection between the host computer 22 and client computing
6 device 24. Once the connection is established (as described below), the system
7 uses known protocols such as PPP, IP, and TCP to exchange data over the serial
8 connection.

9 The host computer 22 has a memory 30, a processor 32, a display 34, one or
10 more input devices 36 (e.g., keyboard, mouse, USB connections, etc.), and
11 multiple communication (COMM) ports 38. The memory 30 generally includes
12 both volatile memory (e.g., RAM) and non-volatile memory (e.g., Flash, ROM,
13 hard disk, etc.).

14 An operating system 50 resides in memory 30 and executes on the processor
15 32. The host computer 22 preferably runs a Windows-brand operating system from
16 Microsoft Corporation, such as Win32-based products (e.g., Windows 95,
17 Windows 98, etc.), although other operating systems may be used. One or more
18 application programs 52 are loaded into memory 30 and run on the operating
19 system 50. Examples of applications 52 include email programs, scheduling
20 programs, word processing programs, Internet browser programs, and so on. The
21 operating system 50 has a set of drivers 54 to manage COMM ports 38, as well as
22 peripheral devices connected through the COMM ports and/or various hardware
23 components.

24 The operating system 50 has a synchronization module 58 to facilitate serial
25 communication between the host computer 22 and the client computing device 24

1 and to synchronize the contents on each machine. The applications 52 call the
2 synchronization module 58 via an API (application programming interface). For
3 instance, a scheduling program on the host computer 22 calls the synchronization
4 module 58 to facilitate the exchange of appointments and tasks between the two
5 computers to bring both computers up to date. As an exemplary implementation,
6 the synchronization module 58 is implemented as part of the Windows CE
7 Services module in the Windows CE operating system.

8 The synchronization module 58 is configured to facilitate serial
9 communication using the Unimodem null serial protocol. According to this
10 protocol, the client computing device 24 initiates a communication session by
11 sending over a message consisting of the text string "C", "L", "I", "E", "N", "T"
12 and the host computer 22 replies with a message "'S", "E", "R", "V", "E", "R",
13 "C", "L", "I", "E", "N", "T'". As part of the configuration, the synchronization
14 module 58 implements a baud rate detection system 60 that automatically detects a
15 baud rate at which the client computing device 24 is transmitting data. The baud
16 rate detection system 60 automatically cycles through a set of possible baud rates
17 and at each rate, listens to detect the message text string "C", "L", "I", "E", "N",
18 "T" from the client computing device 24.

19 The baud rate detection system 60 includes a rate selector 62, a message
20 detector 64, and a rate cache 66. The rate selector 62 selects among multiple baud
21 rates at which the client computing device may transmit the message over the
22 serial connection. The baud rates may be stored, for example, in a table 68.

23 Once a baud rate is selected, the message detector 64 listens to the serial
24 channel at the selected baud rate. If no message is received within a
25 predetermined time period, or only error characters are received (i.e., characters

1 not in the message "C", "L", "I", "E", "N", "T"), the rate selector 62 selects a next
2 baud rate from the table 68 and the message detector 64 listens to the serial
3 channel at this next baud rate. Eventually, the rate selector 62 picks a baud rate
4 that enables the message detector 64 to successfully receive the text string "C",
5 "L", "I", "E", "N", "T". Once this occurs, the rate selector 62 sets the baud rate at
6 the current rate for future communication with the client. The rate selector also
7 caches the baud rate in cache 66 for use in subsequent client connects under the
8 assumption that it is likely that the host computer will communicate again with the
9 same client computing device at the same baud rate. This process is described
10 below in more detail with reference to Fig. 2.

11 With continuing reference to Fig. 1, the client computing device 24 is
12 illustrated as having a processor 70, a display 72, one or more input devices 74
13 (e.g., keypad, touchpad, microphone, etc.), a serial port 76, and a memory 78. The
14 memory 78 represents both volatile and non-volatile memory, and is used to store
15 programs 80 and an operating system 82. As an exemplary implementation, the
16 operating system is the Windows CE operating system from Microsoft
17 Corporation. The Windows CE operating system supports four different baud
18 rates: 19.2K, 38K, 56K, and 115K.

19 Fig. 2 shows a method for automatically detecting a baud rate at which the
20 client computing device 24 is transmitting data. These steps are performed, for
21 example, in software by the baud rate detection system 60 at the host computer 22.

22 At step 100, the baud rate detection system 60 waits for the client
23 computing device to connect. This connection may be a physical connection, in
24 which the client computing device is attached to a serial cable or placed in a
25

1 cradle. The connection may alternatively be non-physical, such as an IR (infrared)
2 connection or an RF (radio frequency) connection.

3 Once a connection is made, the rate selector 62 determines whether there is
4 a baud rate currently cached in rate cache 66 (step 102). If there is (i.e., the “yes”
5 branch from step 102), the rate selector loads the cached baud rate as the current
6 baud rate (step 104). If not (i.e., the “no” branch from step 102), the rate selector
7 62 sets the current baud rate to a default rate identified in the baud rate table 68
8 (step 106).

9 At step 108, the baud rate detection system 60 checks if the client
10 computing device 24 is still connected. If not, it returns to the wait state at step
11 100. Assuming that client is still connected (i.e., the “yes” branch from step 108),
12 the message detector 64 begins listening at the current baud rate for the “C-L-I-E-
13 N-T” text string to be transmitted from the client computing device (step 110).

14 At this point, there are three possibilities: (1) the host receives the correct
15 message; (2) the host receives error characters that are not in the text string “C”,
16 “L”, “I”, “E”, “N”, “T”; or (3) the host receives no characters within a prescribed
17 timeout period. Decision steps 112, 116, and 120 address these three possibilities.
18 At step 112, the message detector 64 determines whether it receives the “C”, “L”,
19 “I”, “E”, “N”, “T” text string. If it does (i.e., the “yes” branch), the host computer
20 22 is using the same baud rate as the client computing device 24 and hence no
21 further adjustment to the baud rate is needed. The rate selector 62 caches the
22 current baud rate into the rate cache 66 for future use during the next connection
23 and the process is completed. At this point, the host computer 22 responds to the
24 client computing device with a response message ““S”, “E”, “R”, “V”, “E”, “R”,
25 “C”, “L”, “I”, “E”, “N”, “T””, which is transmitted at the current baud rate.

1 Suppose, however, that the message detector 64 begins detecting error
2 characters instead of the text string "C", "L", "I", "E", "N", "T" (i.e., the "no"
3 branch from step 112 and the "yes" branch from step 116). Such error characters
4 may occur through aliasing or other artifacts as a result of transmitting at one baud
5 rate and detecting at a different baud rate. In this case, the message detector 64
6 knows immediately that it is using the wrong baud rate and rate selector 64 can
7 select the next baud rate in the baud rate table 68 without waiting for the timeout
8 period to elapse. The process then continues at step 108.

9 Next, suppose that the message detector 64 detects neither the text string
10 ("C", "L", "I", "E", "N", "T") nor error characters (i.e., the "no" branch from step
11 112 and the "no" branch from step 116). In this case, the host computer may or
12 may not be operating at a compatible baud rate. One possible explanation is that
13 the client computing device has not yet sent over the message. Another possible
14 explanation is that the host is indeed operating at an incompatible baud rate. To
15 rule out the former explanation, the message detector 64 listens for a
16 predetermined timeout period, such as 20 seconds. If the message detector 64 fails
17 to detect the text string or error characters within this timeout period (i.e., the "yes"
18 branch from step 120), the rate selector 62 selects the next baud rate in the baud
19 rate table 68 and the process continues at step 108.

20 To provide an example of this process, suppose that the client computing
21 device 24 is set to a baud rate of 19.2K. In addition, suppose that the baud rate
22 table 68 lists four baud rates in the following order: (1) 56K, (2) 19.2K, (3) 115K,
23 and (4) 38K. The rate selector initially selects the default rate of 56K. The
24 message detector 64 listens at 56K, but fails to detect the text string ("C", "L", "I",
25 "E", "N", "T") being transmitted at 19.2K. Accordingly, either upon receiving

1 error characters or upon expiration of the timeout period, the rate selector 62
2 automatically selects the next baud rate of 19.2K. Since this is now the baud rate
3 used by the client computing device 24, the message detector 64 will detect the
4 text string ("C", "L", "I", "E", "N", "T"). The rate selector 62 caches the 19.2K
5 rate in cache 66 and the process is completed.

6 The system and process described above is advantageous because it enables
7 automatic detection of the client baud rate. As a result, connection failures caused
8 by baud rate mismatches are effectively eliminated. This obviates the need for a
9 user to manually reconfigure either the host or client computing device when baud
10 rate mismatches occur.

11 Although the description above uses language that is specific to structural
12 features and/or methodological acts, it is to be understood that the invention
13 defined in the appended claims is not limited to the specific features or acts
14 described. Rather, the specific features and acts are disclosed as exemplary forms
15 of implementing the invention.